Leveraging Agent Properties to Assure Survivability of Distributed Multi-Agent Systems

Marshall Brinn
BBN Technologies
10 Moulton Street
Cambridge, MA 02138
(617)-873-2717

mbrinn@bbn.com

Mark Greaves
DARPA/IXO
3701 North Fairfax Drive
Arlington, VA 22203-1714
(703)-526-6623

mgreaves@darpa.mil

ABSTRACT

The nature of distributed multi-agent systems makes assuring their survivability under stress particularly challenging. However, the nature of distributed agent-based systems also brings the potential to address these particular challenges, and, indeed, to assure survivability to a degree beyond that possible in non-agent-based architectures. This extended abstract synopsizes a paper detailing approaches that are rooted in the essential properties of agent software architectures to assure the survivability of distributed agent-based systems. Specifically, the paper describes efforts under the DARPA UltraLog program to formulate a survivability argument based on properties of agent architectures. This extended abstract truncates many details from the original; interested readers are encouraged to contact the authors for the complete paper.

Categories and Subject Descriptors

C.4 [**Computer Systems Organization**]: Performance of Systems – *Fault Tolerance, Measurement Techniques, Performance attributes, Reliability, availability, and serviceability.*

General Terms

Measurement, Reliability, Security, Verification

Keywords

Survivability, distributed multi-agent systems, agent properties, assurance.

EXTENDED ABSTRACT

Introduction. The development and deployment of distributed multi-agent systems (DMAS) are accompanied by many challenges to the survivability of these systems. The distributed nature of the DMAS make them vulnerable to disruptions to network infrastructure and the unreliability of individual platforms. However, the very nature of agent-based architectures and applications, particularly their autonomous, anonymous nature, provides capabilities that may support and assure survivability of DMAS.

The DARPA UltraLog has provided an opportunity to investigate, enhance and assure survivability of DMAS. [10] The mission of the UltraLog program is to assure the survivability of a particular distributed multi-agent system under very harsh warlike conditions. The results discussed in the paper are derived from the efforts under UltraLog to achieve this assured survivability.

The paper describes a set of capabilities, defenses and design/configuration patterns that constitute a methodology for designing and deploying DMAS that are intrinsically survivable in stressful environments. The elements of this methodology are rooted in several essential properties of agent-based processing. Further, the paper describes efforts to construct a survivability argument for this methodology.

Assured Survivability Definitions. We assume a given system (distributed, agent-based or otherwise) has a given function that it provides, and that the quality-of-service (QoS) for that function can be measured. [7] We define the *survivability* of a system as the extent to which the QoS of that function is maintained under stress. A system is said to *survive* a given stress if the maintained level of QoS is above some threshold defined as minimally acceptable. The *stresses* to which we refer are external events or causes that may serve to degrade the QoS of system function. These will tend to fall into categories of *information attacks, loss of processing resources* and *increased workload.*

Dimensions of DMAS Function. The paper suggests three general categories to span the space of properties of system function: *Absolute* (properties with an absolute character, that is, they may never be violated), *Binary* (properties with a binary character, that is, that they may be present or not present, and should be present as much as possible), and *Partial* (properties with a partial character, that is, they may be wholly, partially or not at all present, and should be as much available as possible).

Properties for Distributed multi-Agent Systems. The approach taken by UltraLog is to devise and deploy defenses and responses to stresses and attacks that enable it to maintain an assured degree of function in the face of these stresses and attacks. Our claim is that in order to construct such defenses and responses we may and indeed should build on particular properties intrinsic to typical distributed agent-based architectures. [5]

UltraLog is built on the Cougaar (Cognitive Agent Architecture) framework [2]. As such, it rests on the general processing and architectural characteristics of Cougaar, including a component-based architecture and agent-internal blackboard construct.

The paper describes a set of properties of distributed agent-architectures from which UltraLog has constructed its defenses against stresses, namely: *Anonymity*, *Public API*, *Mobility*,

AAMAS '03, July 14-18, 2003, Melbourne, Australia.
ACM 1-58113-683-8/03/0007.

Dynamic Capability Discovery, *Autonomy*, *Task Orientation*, and *Composability*.

Methodology for Assurance of Survivability Properties. The paper details different defense strategies against the classes of stresses and attacks. These defenses descr.ued fall into categories of: *Prevent* (place barriers in processing or configuration to deter particular attack or stress types), *Detect* (determine that a particular attack or degradation has occurred), *Contain* (work to keep the degradation associated with a particular attack or stress from affecting other agents or processing tasks) and *Recover* (work to return processing and capabilities to a prior, uncompromised state). The paper discusses that some defenses that can be implemented entirely at the infrastructure layers, while others require cooperation between the infrastructure and application layers.

Assurance Argument Approach. The paper presents, in this section, a general claim for survivability of an UltraLog-based DMAS, an assurance argument supporting this claim, and the evidence needed to support the different sub-assertions made in our argument. Our basic strategy is to first lay out the logical framework of the argument, and derive from this framework a set of properties that of a DMAS that can be measured. Critically, for us, this set of properties is intimately bound into the agent nature of the DMAS. Using this approach, we can show that the characteristic properties of agent technology allow us to quickly build survivability claims for DMAS.

The claim for which we argue is as follows: ***a DMAS built using the above methodology of deployed defenses supports survivability against a given threat model to a predictable degree.*** We argue that in order to achieve survivability of a predictable degree against a given stress model, it is both *necessary* and *sufficient* to construct a given DMAS according to this methodology.

Necessity. The paper describes how, in order for the above claim to hold, it is necessary that defenses against the given stress types be constructed in patterns of Protection, Recovery and Adaptivity.

Sufficiency. The paper argues that in order for a DMAS to exhibit a degree of function against a particular stress model that is predictable, it is sufficient for it to be designed and deployed as described. A general *stress model* is defined as a probability distribution that a given level of hardware availability and a given work load will be present in the system at a given time. We have, in such a model, a nominal tiling of *stress regions* over the resource-workload space and a probability of being in one of the regions. The set of metrics and sensors within a given UltraLog-based DMAS will enable it to determine where it lies in this grid. Adaptive mechanisms allow the system to modify its behaviors to change the realized QoS based on the current stress level. Once the system adopted a particular group of settings across the agents, the system will operate within measurable bounds of QoS. We thus have a particular QoS that can be measured that the system will exhibit in each block in the grid representing a given stress model. The aggregate QoS exhibited by the system against that stress model is given by the probability-weighted sum of the individual QoS in each block, that is,

$$\mathrm{QoS}_{\mathrm{SystemAggregate}} = \sum \mathrm{QoS}_i * P_i$$

Supporting Evidence for the Argument. The paper describes the next steps towards providing evidence that a particular DMAS, in this case the UltraLog prototype, actually exhibits those properties. The evidence that the UltraLog DMAS satisfies our survivability claim falls into three broad classes. First, UltraLog can supply Survivability cases, analogous to traditional Safety cases, that support the claim that all threats in the threat model are covered by the given defenses. Second, UltraLog includes software documentation of the architecture showing the details of each defense and how if implemented correctly it will reliably provide defense and/or recovery from the given class of stresses. Third, we must have the empirical evidence of tests and measurements that the DMAS architectural features are implemented correctly. The UltraLog program performs broad quantitative assessment of the survivability of its specific military logistics DMAS, applying a variety of stresses individually and in aggregate, and measuring the effects on the system via the defined measures-of-performance. These types of evidence are quite lengthy and detailed, and available at [10].

ACKNOWLEDGEMENTS

The work described here was performed under the DARPA UltraLog contract #MDA972-01-C-0025. These ideas represent contributions by many individuals who participated in the DARPA ALP and UltraLog programs. We are particularly grateful to Victoria Stavridou of SRI for her valuable contributions to this effort.

REFERENCES

[1] Bishop, P., Bloomfield, R. Guerra. “A Methodology for Safety Case Development", in Safety-critical Systems Symposium, Birmingham, UK, February 1998

[2] Cougaar Open Source Site. http://www.cougaar.org

[3] Dietrich, S., Ryan, P. “The Survivability of Survivability”. Fourth Information Survivability Workshop, (ISW-2001/2002).

[4] Fisher, D. “Survivability and Simulation”, Third Information Survivability Workshop (ISW-2000).

[5] Huhns, M, Singh, M. editors. Readings in Agents. Morgan Kaufmann, 1996.

[6] Kokar, M., Baclawski, K. Eracar, Y. “Control Theory-based Foundations of Self-Controlling Software”. IEEE Intelligent Systems, May/June, 1999.

[7] Manola, F. Providing System Properties (Ilities) and Quality of Service in Component-based Systems. Object Services and Consulting, 1999

[8] Stafford, J, McGregor, J. “Issues in Predicting the Reliability of Composed Components”. Submitted to 5th ICSE Workshop on Component-based Software Engineering.

[9] Stavridou, V., and Riemenschneider, R.A. "Provably Dependable Software Architectures," in Proc. of the Third ACM SIGPLAN International Software Architecture Workshop, pp. 133-136, 1998.

[10] UltraLog Program Site. http://www.ultralog.net